NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

ENHANCEMENT OF VIDEO IMAGES
DEGRADED BY TURBID WATER

by

Jorge A. Franco

December 1986

Thesis Advisor: Charles W. Therrien

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release;distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 62
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
ENHANCEMENT OF VIDEO IMAGES DEGRADED BY TURBID WATER

12 PERSONAL AUTHOR(S)
Franco, Jorge, A.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1986 December	67

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Image Enhancement, Noise Filtering, Contrast Enhancement

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis deals with the enhancement of video images degraded by turbid water viewing conditions. An algorithm by Peli and Lim has been used with some success for enhancement, but it was found to accentuate noise. The thesis examines a combination of the Peli and Lim algorithm with three - approaches to enhancement.

First, a Short Space Spectral Subtraction algorithm wich performs the restoration in the density domain, using an estimate for the power spectrum of the given data set. The degraded image is divided into many subimages and each subimage is restored separately and then combined.

Next, an algorithm for Image Enhancement by Use of Local Statistics, wich uses the assumption that the sample mean and variance of a pixel is equal to the local mean and variance of all pixels within a fixed range surrounding it.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Charles W. Therrien	22b TELEPHONE (Include Area Code) (408) 646-3347 / 22c OFFICE SYMBOL 62Ti

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted. All other editions are obsolete   SECURITY CLASSIFICATION OF THIS PAGE

(Continuation of block 19)
Combination of the algorithms are applied to degraded images, and the results are compared and discussed, in each case. It was -- found that noise smoothing can be achieved with the spectral subtraction algorithm, and that the local statistics technique yielded very good contrast enhancement.

Approved for public release; distribution is unlimited

Enhancement of Video Images Degraded by Turbid Water

by

Jorge A. Franco
Lieutenant, Colombian Navy
B.S., Escuela Naval "Almirante Padilla", 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1986

ABSTRACT

This thesis deals with the enhancement of video images degraded by turbid water viewing conditions. An algorithm by Peli and Lim has been used with some success for enhancement, but it was found to accentuate noise. The thesis examines a combination of the Peli and Lim algorithm with three approaches to enhancement.

First, a Short Space Spectral Subtraction algorithm which performs the restoration in the density domain, using an estimate for the power spectrum of the given data set. The degraded image is divided into many subimages and each subimage is restored separately and then combined.

Next, an algorithm for Image Enhancement and Noise Filtering by Use of Local Statistics, which uses the assumption that the sample mean and variance of a pixel is equal to the local mean and variance of all pixels within a fixed range surrounding it.

Finally, a median filter for noise reduction , where a given pixel of a degraded image is replaced by the median of the pixel values in a window surrounding it.

Combination of the algorithms are applied to degraded images, and the results are compared and discussed, in each case. It was found that noise smoothing can be achieved with the spectral

subtraction algorithm, and that the local statistics technique yielded very good contrast enhancement.

TABLE OF CONTENTS

I. INTRODUCTION

In this thesis, some algorithms for the enhancement of video images degraded by turbid water viewing conditions are implemented. The images used to illustrate the processes, were recorded during the recovery procedure in a torpedo testing range.

The recovery operation is carried on undersea with the aid of digging equipment and underwater video cameras equipped with strong artificial lights. The equipment is controlled and the operation is monitored on video monitors at the surface onboard the recovery vessel. The recovery equipment in its attempt to dig out the torpedo, stirs up sediment wich visually obscures the object of interest and impedes the operation. To date no special techniques have been used operationally to process or enhance the video image before display.

When it is of interest to "improve" an image, there are two broad types of image manipulation processes that cover all operations performed to get such improvement: image restoration and image enhancement.

The goal of image enhancement is to process a degraded image so that the result is more suitable than the original image for a specific application or aids the human analyst in the extraction and interpretation of pictorial information [Ref. 1]. Image restoration, on the other hand, is intended to bring a degraded

image back to an ideal degradation-free image as closely as possible [Ref. 2]. This makes the image enhancement problem a subjective one; the improvement of the image appearance to the human viewer is highly dependent on the viewer himself and on the application. What is "good" for one person or application is not necessarily "good" for another one. Consequently, the suitability of the processed image for a specific application makes image enhancement techniques very much problem-oriented.

The degradation process can result from causes such as, an imperfect photographic process, imperfect display devices, poor contrast due to environmental conditions, different forms of noise (channel, quantization, salt and pepper) and others [Ref. 3]. In the same manner, techniques for enhancement are dependent of the original degradation process. Processes yielding satisfactory results for one type of degradation are not necessarily suitable for another.

The interest of this thesis is on examining different techniques to enhance images that are degraded by environmental conditions that create a great lack of contrast in the recorded images. This work is a continuation of previous research begun on the subject, which included the implementation of an adaptive filtering [Ref. 4] algorithm for contrast enhancement. This algorithm was applied to a set of underwater images recorded in turbid water viewing conditions. The images are low contrast degraded by noisy background. As will be explained in the next chapter, the

algorithm yielded very good contrast enhancement but, seemed to accentuate the noise. The purpose of this work is to investigate some noise smoothing techniques to be applied to the contrast-enhanced images, an alternative contrast manipulation scheme, and to compare the results obtained to those found with the algorithm already implemented. Besides the images included in the experimental results of this thesis, many more images resulting from other combinations and variations of the processes and different settings of the parameters were obtained. We have included here only those thought to be most representative of the results obtained.

II. OVERVIEW OF PREVIOUS WORK

A. DESCRIPTION OF THE ADAPTIVE FILTERING ALGORITHM

An adaptive filtering algorithm for image enhancement by Peli and Lim [Ref. 4] has been used for contrast enhancement of images. The images are typifyed by a large dynamic range, and recorded over a medium, characterized by a much smaller dynamic range, which causes those regions of the images with very high or very low luminance to be poorly represented. This algorithm modifies the local luminance mean of an image and controls the local contrast as a function of the local luminance mean of the image.

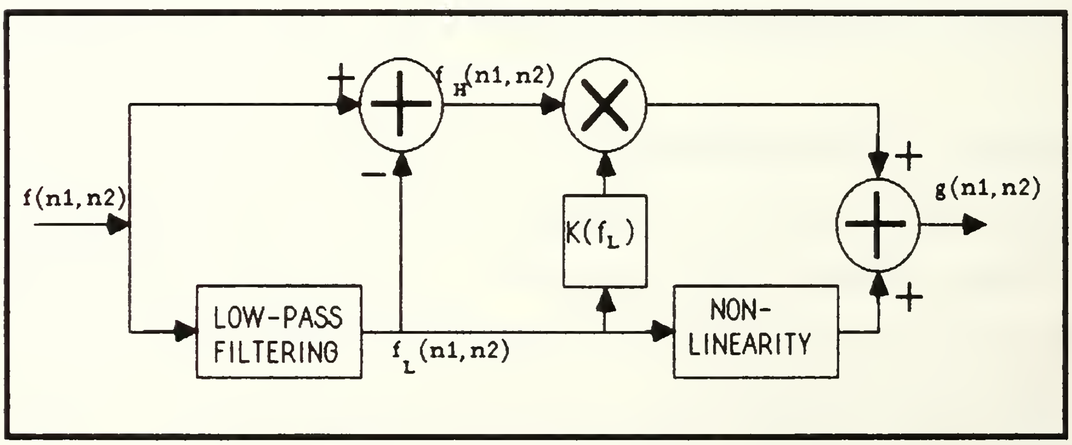


Figure 2.1. Adaptive Filtering for Image Enhancement

In Figure 2.1 a block diagram of the Adaptive Filtering algorithm is shown . In the figure, $f(n_1, n_2)$ denotes the

unprocessed digital image, and $f_L(n_1,n_2)$, which denotes the local luminance mean of $f(n_1,n_2)$, is obtained by low-pass filtering $f(n_1,n_2)$. The sequence $f_H(n_1,n_2)$ which denotes the local contrast is obtained by subtracting $f_L(n_1,n_2)$ from $f(n_1,n_2)$. The local contrast is modified by multiplying $f_H(n_1,n_2)$ with $k(f_L)$, a scalar which is a function of $f_L(n_1,n_2)$. The modified contrast is denoted by $f'_H(n_1,n_2)$. The specific functional form of $k(f_L)$ depends on the particular application under consideration; a value of $k(f_L) > 1$ represents local contrast increase while $k(f_L) < 1$ represents local contrast decrease. The local luminance mean is modified by a point non-linearity and the modified local luminance mean is denoted by $f'_L(n_1,n_2)$. The specific non-linearity chosen depends on the particular application under consideration, and in most application problems the non-linearity is chosen so that the overall dynamic range of the resulting image is approximately the same as the dynamic range of the recording medium. The modified local contrast and local luminance mean are then combined to obtain $g(n_1,n_2)$, the processed image.

B. DISCUSSION OF RESULTS

In this section the application of the algorithm described in the previous section to the images shown in Figure 2.2 is presented. Figure 2.2 shows two low-contrast noise-degraded images of 512 × 512 pixels with each pixel represented by 8 bits. Note that the small dynamic range of the luminance of both images make

(a)

(b)

Figure 2.2 Original Degraded Images.

difficult the appreciation of subtle details in them. In this case, it is desired to increase the local contrast in the images to bring back the lost details. To achieve this, $k(f_L)$ is chosen as shown in Figure 2.3 (a) and (b) for the images of Figure 2.2(a) and (b), respectively. For the case of Figure 2.2(a), the function is chosen to increase the local contrast for the low luminance regions, maintaining it relatively constant for medium ranges of local luminance, and again enhance it, for the brighter regions of the degraded image. The results are shown in Figure 2.4(a), where a relative contrast increase can be observed. This is especially evident in the "fish" shape close to the lower right corner of the image. Also note the increase in background noise resulting in the textured background observed in the contrast-enhanced image.

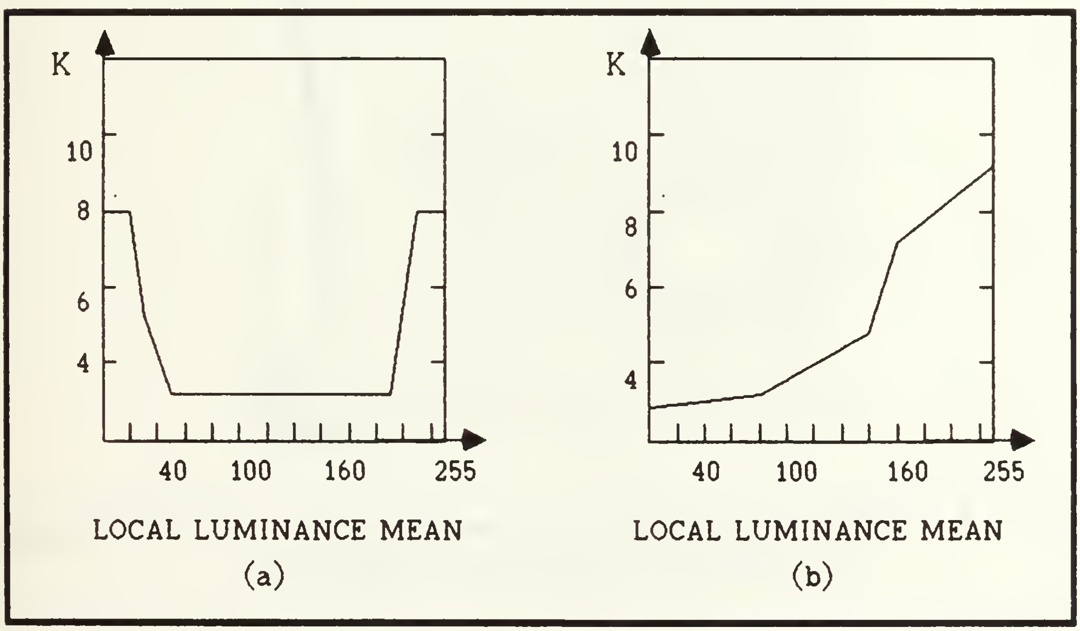
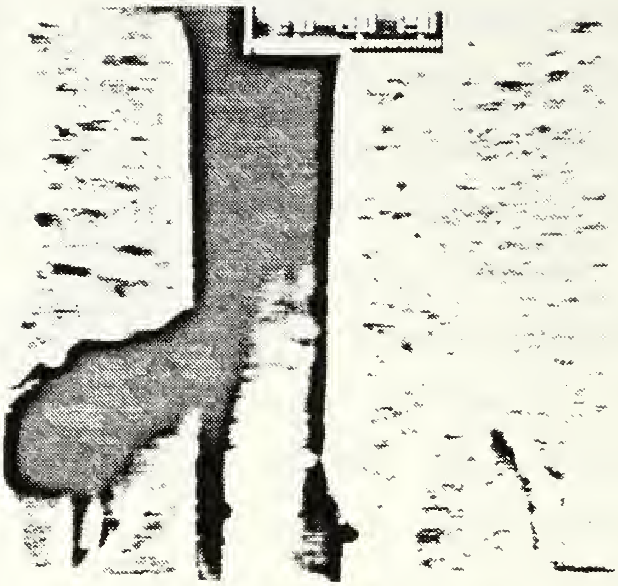


Figure 2.3. $K(f_L)$ Function for Processing Images in Figure 2.2



(a)

(b)

Figure 2.4. Images in Figure 2.2 Processed with Adaptive Filtering for Image Enhancement Algorithm.

In the case of the image of Figure 2.2(b), it is desired to retain the original local contrast in the darker regions, but enhance it in the regions corresponding to medium and high local luminance means. This explains the particular shape of the $k(f_L)$ function chosen to process this image. In the results, shown in Figure 2.4(b), some details difficult to be distinguished in the un-processed image, are now more evident. Among them are the particular shape of the object near the center of the image and the form of the ocean floor below the left-most object. Again, it can be observed that there is an increase in the background noise throughout the contrast-enhanced image.

III. DESCRIPTION OF THE ALGORITHMS

In this chapter, the theorethical foundations and procedures for implementation of the different algorithms used in the noise-filtering and contrast enhancement problem are presented. We start with a description of noise-smoothing techniques to be used as post-processing after the adaptive filtering algorithm. Then an alternative method for the contrast enhancement is presented. Finally, the use of median filtering for noise-smoothing is described.

A. SHORT SPACE SPECTRAL SUBTRACTION

This algorithm, developed by J. S. Lim [Ref. 5] performs the restoration in the frequency domain, using a converging solution for the power spectrum of a given data set that does not depend on the original estimate. The procedure is based on two basic assumptions. First, each part of an image generally differs sufficiently from other parts so that the image cannot be modeled by a stationary random process. Second, the power spectrum of the restored image is estimated by the spectral subtraction of the additive noise spectra from the degraded image.

1. Spectral Subtraction

Frequency domain techniques for image enhancement use iterative procedures for estimating the image power spectral

density, continuing until certain fidelity criteria are met. The density so obtained, is dependent on both the initial estimate for starting the iterative procedure and the stopping criterion. Some methods converge more rapidly than others, and sometimes first and second order statistics of the degrading function must be known.

Consider a model of image degradation as presented in Figure 3.1. In this model $f(n_1,n_2)$ represents a digital image, $b(n_1,n_2)$ represents a linear space-invariant point spread function, and $d(n_1,n_2)$ represents an additive noise component. Thus, $g(n_1,n_2)$ represents a blurred version of $f(n_1,n_2)$ and $y(n_1,n_2)$ is the degraded image.

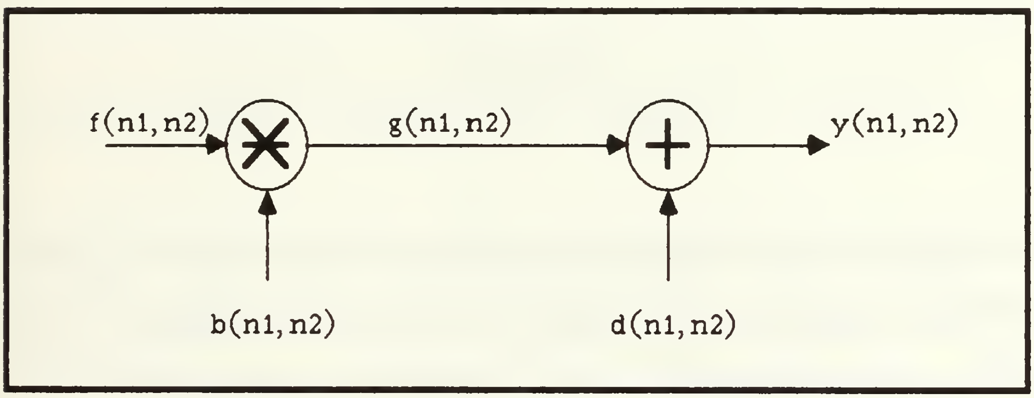


Figure 3.1. Model of Image Degradation

An image restoration system is shown in Figure 3.2. It can be seen that the goal of a restoration system is to process a degraded image $y(n_1,n_2)$ through a noise reduction system to estimate $g(n_1,n_2)$ which in turn is processed by a deblurring

system to get $f(n_1,n_2)$, an estimate of the original blur-free noise-free image.

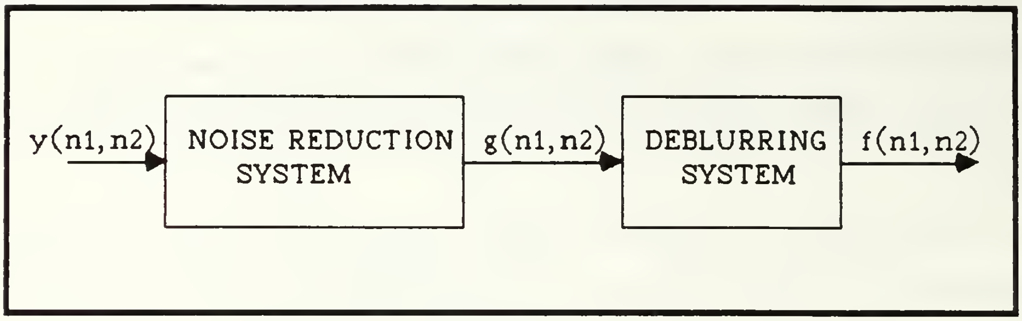


Figure 3.2. Image Restoration System

The purpose of this algorithm is to provide a model for the Noise Reduction System, as presented in the Figure. A general model for the restoration filter is given by

$$H(\omega_1,\omega_2) = \left\{ \frac{P_g(\omega_1,\omega_2)}{P_g(\omega_1,\omega_2) + \alpha \cdot P_d(\omega_1,\omega_2)} \right\}^{\beta} \qquad (3.1)$$

where $P_g(\omega_1,\omega_2)$ and $P_d(\omega_1,\omega_2)$ represent the power spectra of $g(n_1,n_2)$ and $d(n_1,n_2)$ respectively, α and β are constants. If β is unity, $H(\omega_1,\omega_2)$ corresponds to the parametric Wiener filter, and if α also is unity, it reduces to the standard Wiener filter [Ref. 1,2]. Other values for α and β define different filtering techniques, such as power spectrum filtering and geometrical mean filtering. One commonly used approach in the implementation of $H(\omega_1,\omega_2)$ is an iterative procedure which begins with an initial estimate of $P_g(\omega_1,\omega_2)$ and then iteratively estimates $g(n_1,n_2)$ and $P_g(\omega_1,\omega_2)$

until a converging solution or a desirable performance is achieved. This iterative procedure may be computationally undesirable.

The Spatial Subtraction approach analytically obtains a converging solution, by estimating $P_g(\omega_1,\omega_2)$ from $g(n_1,n_2)$ as $(1/k)\cdot|G(\omega_1,\omega_2)|^2$ and using a value for β of ½. The complete solution for this problem, is given by

$$|\hat{G}(\omega_1,\omega_2)| = \sqrt{|Y(\omega_1,\omega_2)|^2 - \alpha\cdot k\cdot P_d(\omega_1,\omega_2)} \quad (3.2)$$

$$\measuredangle\, \hat{G}(\omega_1,\omega_2) = \measuredangle\, Y(\omega_1,\omega_2) \quad (3.3)$$

for $\quad |Y(\omega_1,\omega_2)|^2 \geq \alpha\cdot k\cdot P_d(\omega_1,\omega_2)$

and zero otherwise

Here $G(\omega_1,\omega_2)$ represents the discrete space Fourier Transform of $g(n_1,n_2)$, $Y(\omega_1,\omega_2)$ is the discrete space Fourier Transform of $y(n_1,n_2)$, $P_d(\omega_1,\omega_2)$ represents the power spectrum of $d(n_1,n_2)$, the additive noise component of the degradation model, and k is a scaling factor that normalizes the power and energy spectral densities. The phase of $g(n_1,n_2)$ is estimated by the phase of $y(n_1,n_2)$ and the transform magnitude of $g(n_1,n_2)$ is estimated by a particular form of spectral subtraction.

2. Short Space Implementation

As mentioned before, the basic assumption is that each part of an image $f(n_1, n_2)$ generally differs sufficiently from other parts so that it cannot be modelled by a stationary random field. Thus a short space implementation seems appropiate. The technique consists of dividing the original image into many subimages and restoring each subimage separately. The procedure involves the application of a short space window function $w_{i,j}(n_1, n_2)$ to overlapping portions of the degraded image $y(n_1, n_2)$ so that:

$$y(n_1, n_2) \cdot w_{i,j}(n_1, n_2) = \{g(n_1, n_2) + d(n_1, n_2)\} \cdot w_{i,j}(n_1, n_2) \quad (3.4)$$

or, equivalently,

$$y_{i,j}(n_1, n_2) = g_{i,j}(n_1, n_2) + d_{i,j}(n_1, n_2) \quad (3.5)$$

The noise reduction system is then applied to $y_{i,j}(n_1, n_2)$ to recover $g_{i,j}(n_1, n_2)$. The overall full-size restored image $g(n_1, n_2)$ is given by

$$g(n_1, n_2) = \sum_{i=0}^{2K} \sum_{j=0}^{2L} g_{i,j}(n_1, n_2) \quad (3.6)$$

The window function is desired to be a smooth function to avoid some possible discontinuities that may appear at the subimage boundaries in the processed image and must satisfy:

$$\sum_{i=0}^{2K} \sum_{j=0}^{2K} w_{i,j}(n_1, n_2) = 1 \qquad (3.7)$$

for all n_1, n_2 of interest

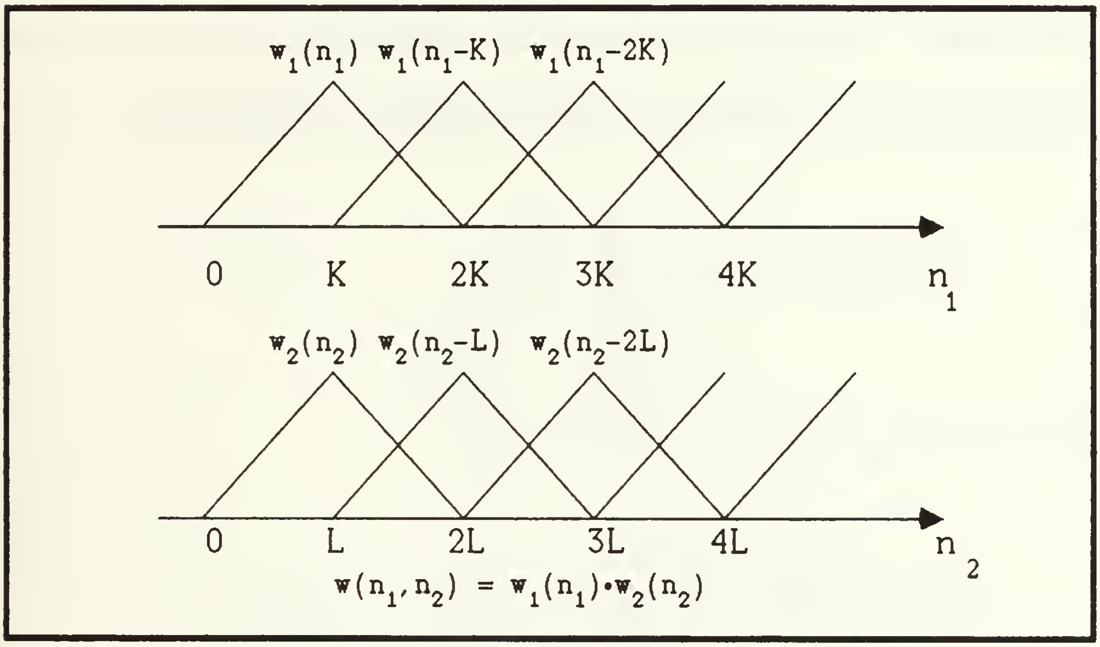


Figure 3.3. 2-D Separable Triangular Window

The condition in Equation 3.6 guarantees that an image can be reconstructed from its subimages. To ensure the smoothness of the window, a 2-D separable triangular window similar to that shown in Figure 3.3 will be used. Each window overlaps its neighboring window by half the window length in each dimension.

B. CONTRAST ENHANCEMENT BY USE OF LOCAL STATISTICS

This algorithm presents a computational technique for contrast enhancement on a two-dimensional image array based on the local mean and variance. The algorithm, as developed by J. Lee [Ref. 6] uses the basic assumption that the sample mean and variance of a pixel is equal to the local mean and variance of all pixels within a fixed range surrounding it.

Let $x_{i,j}$ be the brightness of a pixel (i,j) in a two dimensional N×N image. The local mean and variance are calculated over a $(2n+1)\times(2m+1)$ window. The local mean is defined as

$$m_{i,j} = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{i+n} \sum_{l=j-m}^{j+m} x_{k,l} \qquad (3.8)$$

and the local variance is defined as

$$v_{i,j} = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{i+n} \sum_{l=j-m}^{j+m} (x_{k,l} - m_{i,j})^2 \qquad (3.9)$$

The algorithm is designed such that a pixel $x_{i,j}$ will maintain its local mean, and yet permit its variance to be modified by a constant factor times its original variance. The procedure is defined by

$$x'_{i,j} = m_{i,j} + k(x_{i,j} - m_{i,j}) \qquad (3.10)$$

where k, the gain, is the ratio of new local standard deviation to the original standard deviation. This approach has the computational advantage that the local variance $v_{i,j}$ is not required and only the local mean $m_{i,j}$ needs to be computed. If we have $k > 1$, the image will be sharpened as if acted upon by a high-pass filter. If $0 \le k < 1$, the image will be smoothed as if passed through a low-pass filter. In the extreme case, one has $k = 0$ and $x'_{i,j}$ is equal to its local mean $m_{i,j}$.

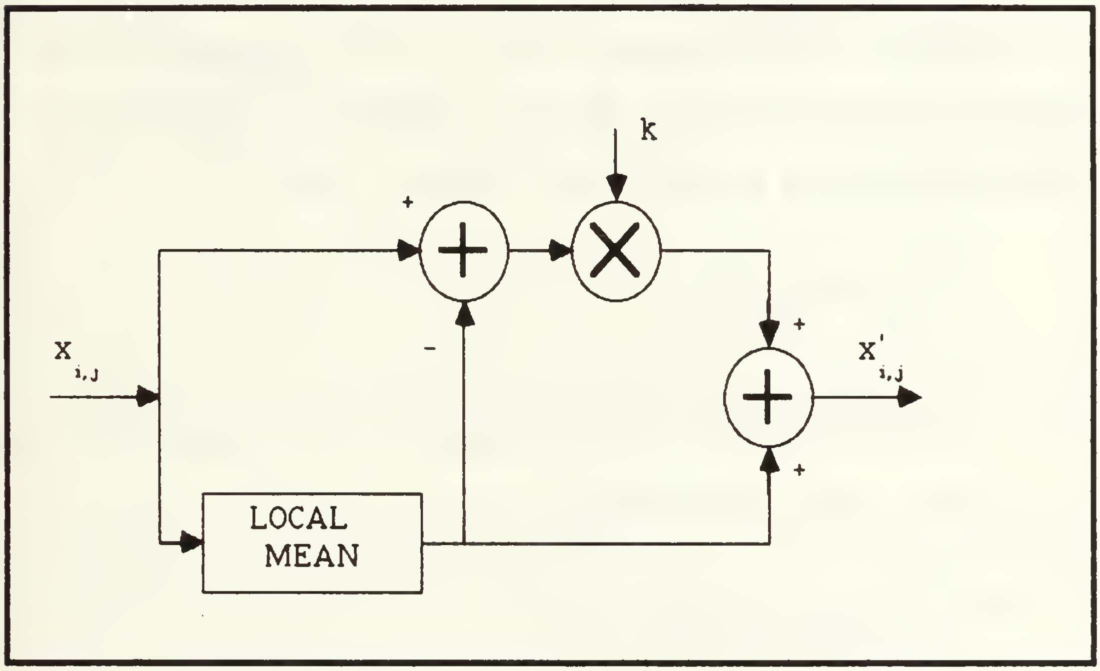


Figure 3.4. Contrast Enhancement by Use of Local Statistics

Figure 3.4 shows a block diagram of the algorithm. The local mean is evaluated over a (2n+1)(2m+1) window and subtracted from the original brightness of the pixel, $x_{i,j}$. The difference is

then multiplied by the standard deviation ratio to enhance the local contrast. Finally, this result is added to the local mean yielding the reconstructed pixel value $x'_{i,j}$.

C. NOISE FILTERING BY USE OF LOCAL STATISTICS

This algorithm, also developed by J. Lee [Ref. 6] is actually a variation of the previous algorithm adapted for the noise filtering problem. In this algorithm, the *a priori* mean (or variance) of the estimated image is calculated as the difference between the mean (or variance) of the noise corrupted image and the mean (or variance) of the noise itself. Let $z_{i,j}$ be the degraded pixel $x_{i,j}$. Then the degraded pixel can be modelled as the sum of the noise-free pixel and a white noise sequence. That is,

$$z_{i,j} = x_{i,j} + w_{i,j} \tag{3.11}$$

where $w_{i,j}$ is the white random sequence with $E[w_{i,j}]=0$. Define the *a priori* mean and variance of $x_{i,j}$ to be

$$\bar{x}_{i,j} \triangleq E[x_{i,j}] = E[z_{i,j}] = \bar{z}_{i,j} \tag{3.12}$$

and

$$Q_{i,j} \triangleq E[(x_{i,j}-\bar{x}_{i,j})^2] = E[(z_{i,j}-\bar{z}_{i,j})^2 - \sigma_1^2 \tag{3.13}$$

respectively. The estimated pixel value $x_{i,j}$ is computed by

$$\hat{x}_{i,j} = \bar{x}_{i,j} + k_{i,j}(z_{i,j} - \bar{x}_{i,j}) \qquad (3.14)$$

where the gain is given by

$$k_{i,j} = \frac{Q_{i,j}}{Q_{i,j} + \sigma_1^2} \qquad (3.15)$$

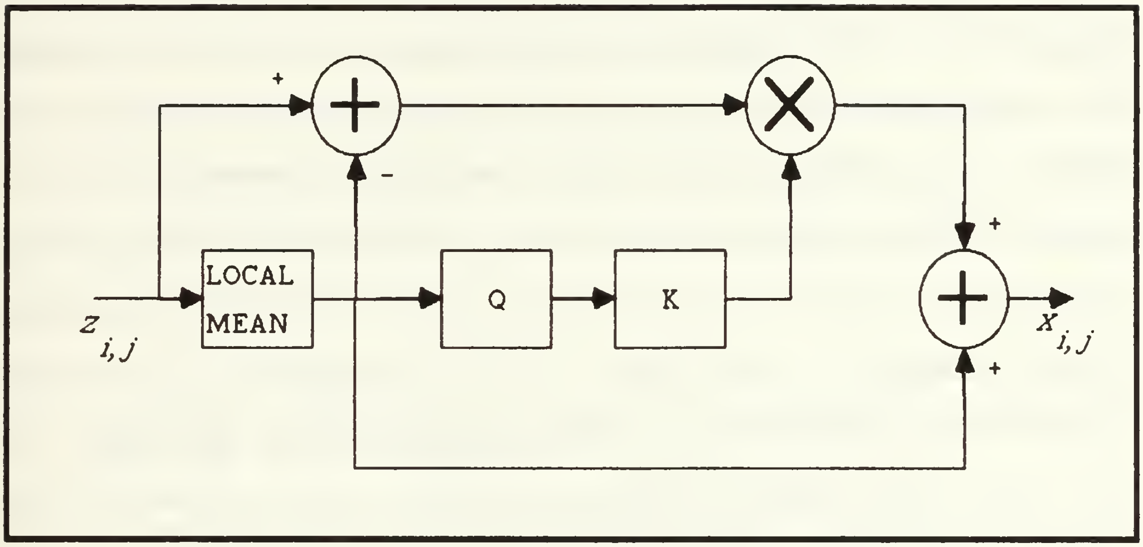


Figure 3.5. Noise Filtering by Use of Local Statistics

Figure 3.5 shows a block diagram for the implementation of the algorithm. Starting with the degraded pixel $z_{i,j}$, the local mean is evaluated and subtracted to obtain the local contrast. Then, both $Q_{i,j}$ and $k_{i,j}$ can be evaluated, and the value of k thus obtained is multiplied by the local contrast. This in turn is added to the local mean to get the restored pixel $x_{i,j}$.

One important parameter is the size of the window over which the local mean is estimated. If the window is too small, the

noise filtering algorithm is not effective. If the window is too large, subtle details of the image will be lost in the filtering process.

D. MEDIAN FILTERING

Median Filtering is a non-linear signal processing technique originally developed by J.Tukey [Ref. 7], that is useful for noise suppression in images. In one-dimensional form the median filter consists of a sliding window encompassing an odd number of pixels. The center pixel in the window is replaced by the median of the pixels in the window. The median of a discrete sequence $a_1, a_2, \ldots, a_N$ for M odd is that member of the sequence for which $(M-1)/2$ elements are smaller or equal in value, and $(M-1)/2$ elements are larger or equal in value [Ref. 2]. For example, if the ordered values of the pixels within a window are 80, 90, 200, 110, 120, the center pixel would be replaced by the value 110, which is the middle point of the sorted sequence 80, 90, 110, 120, 200.

The concept of median filter is easily extended to two dimensions by utilizing a two-dimensional window of some desired shape such as a rectangle or a discrete approximation to a circle. A two-dimensional LXL median filter will provide a greater degree of noise suppression than sequential horizontal and vertical processing with LX1 median filters, but two-dimensional processing also results in greater signal suppression.

The median filter has been found to be more effective than a linear filter for smoothing images with spiky noise degradations

because of extrema rejection by the median [Ref. 8]. Furthermore, the median filter preserves monotonic step edges, that is, it does not blur sharp edges as a linear low-pass filter would.

Another interesting aspect of median filtering that has been studied [Ref. 9], is that of the convergence of a filtered image to what has been called a *root signal*. A *root* is a signal which is invariant under filtering by a particular median filter. The technique consists in making succesive passes of a noise-degraded image thru a median filter until an image corresponding to a root signal is achieved. Definition of a root signal follows, where a window width of 2N+1 is used. This definition uses the following ideas:

1) A *constant neighborhood* is a region of at least N+1 consecutive identically valued samples.

2) An *edge* is an increasing or decreasing sequence of samples which is immediately preceded and followed by constant neighborhoods. An edge cannot contain any constant neighborhood.

3) An *impulse* is a sequence of at most N consecutive samples whose values are different from those of the two surrounding regions; the two surrounding regions are identically valued constant neighborhoods.

A signal is a *root* if and only if it contains only edges alternating with constant neighborhoods [Ref. 10]. The median filtering preserves edges and constant neighborhoods but eliminates impulses. The technique is based on the idea that passing a noise corrupted root signal through the median filter a sufficient number

of times will produce another root signal, which is usually "close" to the original root signal. In reference [Ref. 10], the concept of "closeness" and a measure of the number of runs to be used are calculated as a function of the image and window sizes.

In this thesis, a median filter included in the Spider library [Ref. 11] is used, which is the implementation of the algorithm suggested by Huang, Yang and Tang [Ref. 12]. It consists primarily of an efficient way of updating the histogram each time that a given pixel is replaced by the median within a window. The algorithm is described briefly for a 3X3 window:

Step 1: Obtain the histogram in the first window [Figure 3.6 (a)] and find the median MDN. Next, count the pixels with values smaller than the median of the window. LTMDN is that number.

Step 2: Shift the window right by one pixel [Figure 3.6(b)] and update the histogram and LTMDN. First decrement the counted values of the histogram equivalent to gray level values ($gl(a), gl(d), gl(g)$) of pixels a, d and g. That is:

$$\text{HIST}[gl(a)] = \text{HIST}[gl(a)]-1$$
$$\text{HIST}[gl(d)] = \text{HIST}[gl(d)]-1$$
$$\text{HIST}[gl(g)] = \text{HIST}[gl(g)]-1$$

LTMDN is updated as follows:

IF $gl(a)$ < MDN , LTMDN = LTMDN-1

IF $gl(d)$ < MDN , LTMDN = LTMDN-1

IF $gl(g)$ < MDN , LTMDN = LTMDN−1

In the same way for pixels c', f', and i',

HIST[$gl(c')$] = HIST[$gl(c')$]+1

HIST[$gl(f')$] = HIST[$gl(f')$]+1

HIST[$gl(i')$] = HIST[$gl(i')$]+1

IF $gl(c')$ < MDN , LTMDN = LTMDN+1

IF $gl(f')$ < MDN , LTMDN = LTMDN+1

IF $gl(i')$ < MDN , LTMDN = LTMDN+1

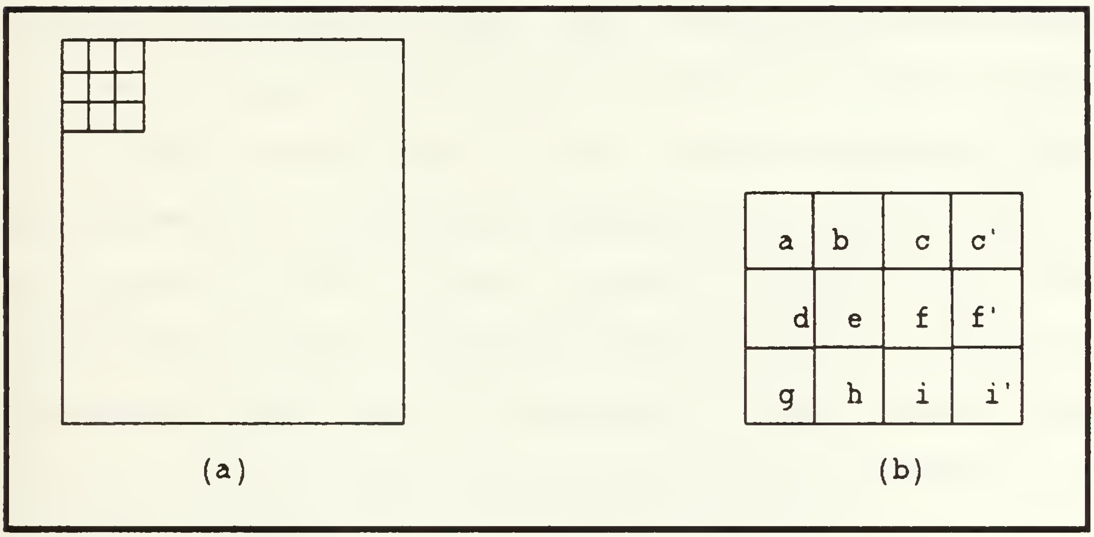


Figure 3.6. Window for Median Filtering

Step 3: Update MDN in the previous window, and obtain a median in the new window. Let ITH=(number of pixels in the window)/2

- If LTMDN > ITH, then

  LTMDN = LTMDN − HIST[MDN]

MDN = MDN−1

Repeat this until LTMDN ≤ ITH is obtained.

- If LTMDN ≤ ITH, then

MDN = MDN+1

LTMDN = LTMDN + HIST[MDN]

Repeat this until LTMDN + HIST[MDN+1] > ITH is obtained. This MDN is the median for the current window and substitute it in the output image.

Step 4: End when one line is finished. Go to Step 2.

IV. EXPERIMENTAL RESULTS

In this chapter, experimental results for different combinations of the algorithms described are presented and discussed. Basically, the adaptive filtering algorithm [Ref. 4] was used for contrast enhancement, as described in Chapter II. As previously stated, this algorithm yielded very good contrast enhancement, but also tended to accentuate the noise. Thus, methods for noise reduction, such as short space spatial subtraction [Ref. 5], noise filtering by use of local statistics [Ref. 6], and median filtering are used, as post-processors to the contrast enhancement operation. The results are shown in next sections. Another technique for contrast manipulation, contrast enhancement by use of local statistics [Ref. 6] was also used, as an alternative to the adaptive filtering algorithm. These results are also given and compared to those previously obtained.

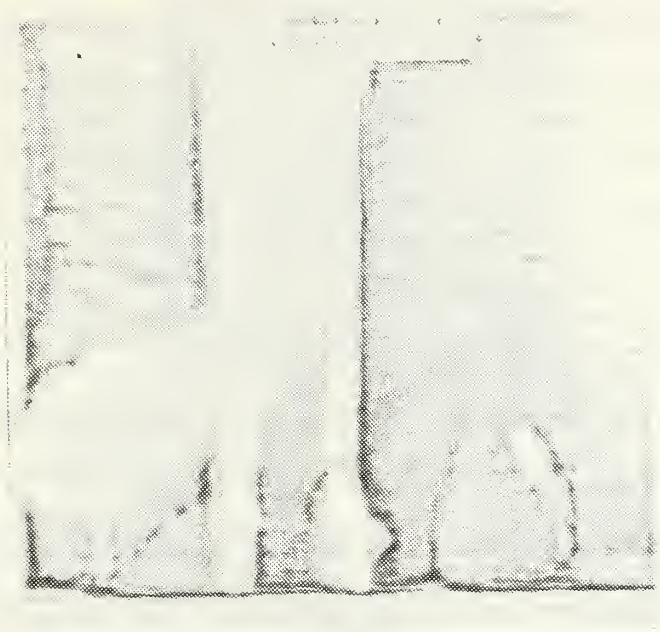
A. PROCESS TYPE 1

This section shows a combination of the adaptive filtering and short space spatial subtraction algorithms. In Figure 2.2 two degraded images are shown. The images are characterized by poor contrast due to the turbid water viewing conditions. These images were processed for contrast enhancement using the adaptive

filtering technique, but the noisy background was heavily accentuated as shown in Figure 2.4. Then the spectral subtraction algorithm was used to smooth the noise resulting in the images shown in Figure 4.1. Inspection of these images shows that this particular algorithm was succesful in smoothing the background noise present in the contrast-enhanced images, and, at the same time, most of the information is preserved. Particularly note, in Figure 4.1(a), how details in the beam-shaped object in the center of the image are brought back, as well as those of the "fish" in the lower right corner of the image. The same observation can be made about Figure 4.1(b), in which the noise reduction was achieved, at the cost of some signal degradation, in this case.

This algorithm requires the power spectral density of the noise to be estimated. In this case, this was accomplished by taking a portion of the noisy background, without any objects and using it to estimate the spectrum. Figure 4.2 shows a representation of the estimated spectrum of the noise density thus obtained. In the figure, the magnitude of the power spectrum is represented as the intensity of a 2-D image, each of the image dimensions being the corresponding spatial frequency coordinates.

For the short space implementation, it is required to process the degraded image by dividing it into subimages and processing each subimage separately. In dividing the degraded image, the size of the subimage must be such that the image luminance in it can be approximated by a stationary random process. In addition, the



(a)

(b)

Figure 4.1. Images in Figure 2.4 Processed with the Short Space Spatial Subtraction Algorithm

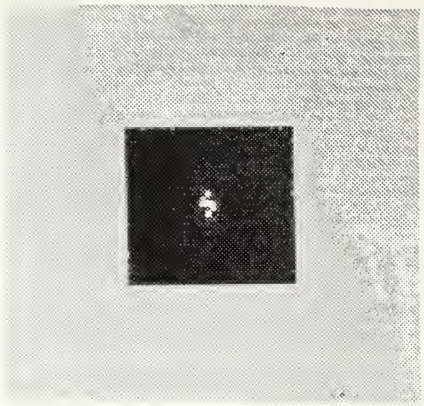


Figure 4.2. Power Spectrum of the Noise in Figure 2.4(a)

window function must be a smooth function in order to avoid possible discontinuities that may appear at the subimage boundaries in the processed image. Thus, two types of smooth functions were used- separable 2-D triangular and Hanning windows-of size 32X32 pixels, overlapped with its neighboring window by half the window duration in each dimension. It was noted that there was no appreciable difference in using either window -triangular or Hanning.

The parameter k, appearing in Equation 3.1 is a scaling factor that normalizes the power spectral densities, and was evaluated as

$$k = \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} w^2(l_1, l_2) \qquad (4.1)$$

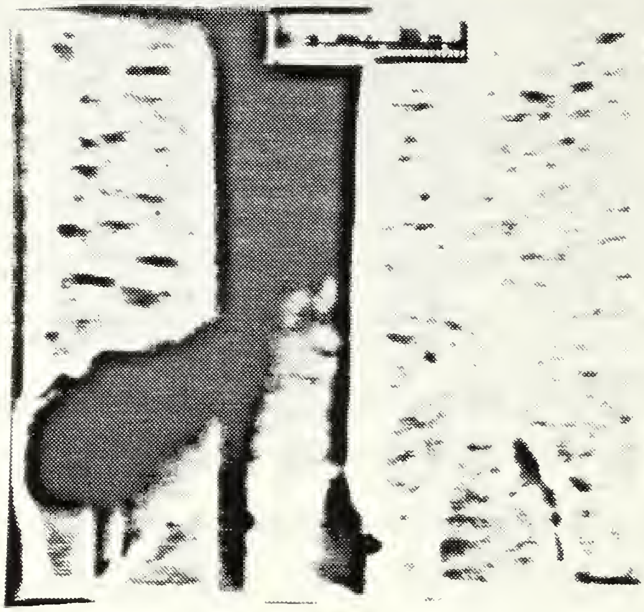
where $w(l_1, l_2)$ is the window function used.

B. PROCESS TYPE 2

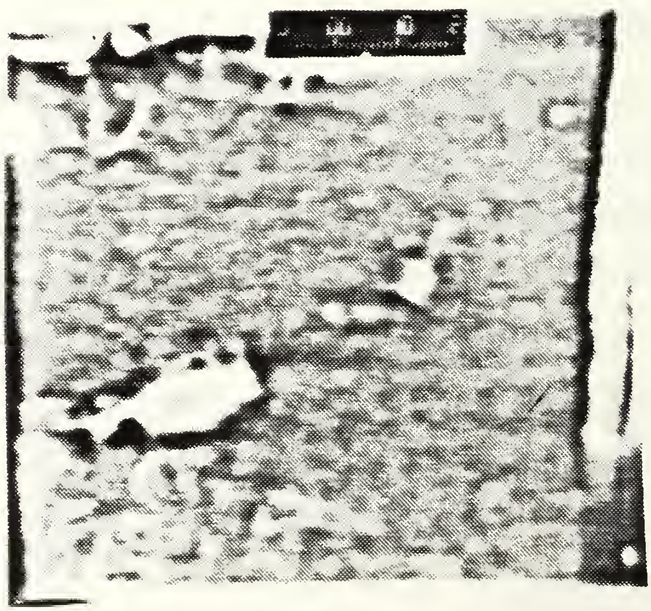
In this section, a combination of the adaptive filtering for Image enhancement and noise filtering by use of local statistics is presented. The images in Figure 2.4 were processed with the noise filtering algorithm, using different values for the noise variance. Based in the data used for generating the noise power spectrum shown in Figure 4.1, a noise variance of 60 was estimated, for the image in Figure 2.4(a). For the image in Figure 2.4(b), a noise variance of 45 was estimated. The algorithm was found to be quite sensitive to the size of the window used to evaluate the local statistics, the a posteriori mean and variance. Window sizes of 8X8, 16X16, and 24X24 pixels were used without noticing any appreciable improvement in the processed image. Figure 4.3 shows the results of processing the images in Figure 2.4 using a window size of 32X32 and the noise variances mentioned above. Note that a relative reduction in the background noise was achieved, with the processed image resembling the effect of low-pass filtering in the background, but without blurring the edges of the objects in the images. When larger sizes for the window were used, some distortion in the images begun to appear.

C. PROCESS TYPE 3

In this section, the result of applying a median filter to a contrast-enhanced image, using the adaptive filtering algorithm, is presented. The image in Figure 2.4(a) was processed with the



(a)



(b)

Figure 4.3. Images in Figure 2.4 Processed with Noise Filtering by Use of Local Statistics Algorithm

median filter available in the Spider Library, using a window size of 5×5, and making succesive iterations of the processed image, until appreciable image degradation start to show up. The image obtained after 8 iterations is shown in Figure 4.4. It can be observed that some noise smoothing was obtained. Even when the resulting image is not noise-free, the remaining noise is of much lower spatial frequency than that of the image before the median-filtering, and thus seems less objectionable.

Detailed characteristics of the signal are preserved and some details have actually been enhanced. This is true in the case of the fish in the lower right corner of the image. A comparison with the use of 3X3 window for the filter operation led to the conclusion that the smaller window size gives slightly more fidelity in the signal preservation. However, since convergence for the 3X3 window is much slower, this choice is undesirable. A reduction in the dynamic range of the median-filtered image can also be observed, which presents an effect similar to that of low-pass filtering in the background region of the image. However, this procedure does not have the undesirable characteristic tendency of the low-pass process to blur the edges of the signal.

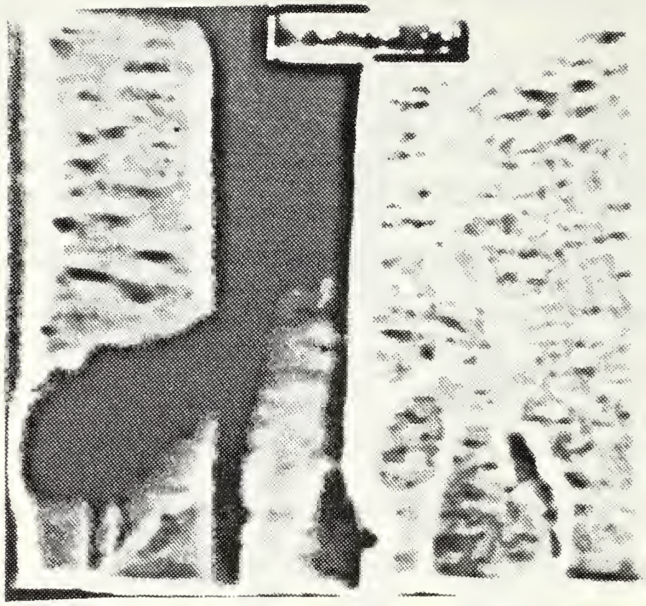


Figure 4.4. Median Filtering of Image in Figure 2.4(a)

D. PROCESS TYPE 4

In this section the application of the contrast enhancement by use of local statistics algorithm is presented. In Figures 4.5 and 4.6 four low-contrast noise-degraded images are shown. All of them are 512 × 512 pixels in size and each pixel is represented by 8 bits. The images in Figures 4.5 (a),(b) and 4.6(a) are characterized by a relatively dark background with a diffused effect around the objects in the pictures resulting in low contrast. The image in Figure 4.6(b) has an almost uniform brightness throughout the picture, which makes it difficult to distinguish between the fish and beam shapes from the uniform background.

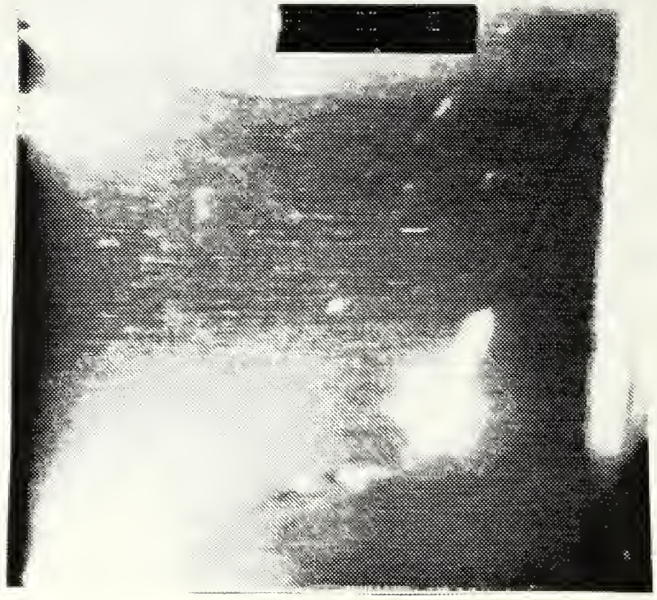
Figure 4.7(a) corresponds to the image in Figure 4.5(a) processed with the contrast enhancement algorithm, using a value

of 4 for the gain k of Equation 3.10. (Recall that k is the ratio of the local standard deviation of the processed image to the original standard deviation). Note the enhancement in contrast achieved, which is particularly evident in the cable connecting the two objects in the lower part of the image.

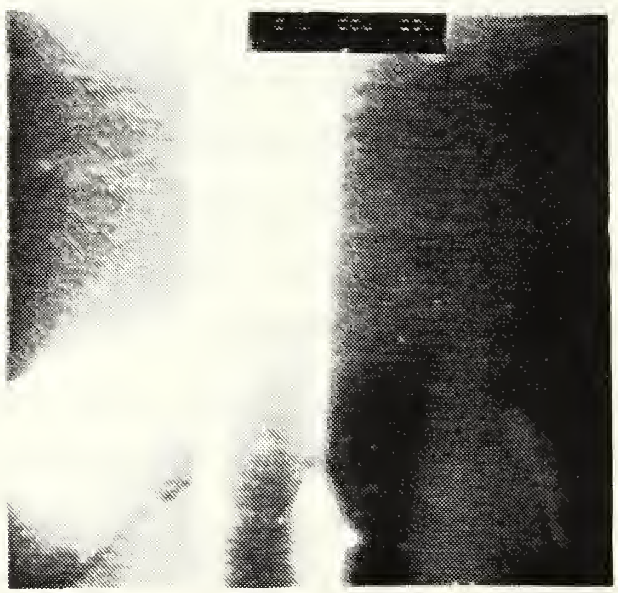
The image in Figure 4.5(b) was processed using a value of k=6, with results shown in Figure 4.7(b). The higher value was used to try to enhance the fish shape in the lower right part of the image. This enhancement was achieved to some extent.

Figure 4.8(a) shows the result of processing the image in Figure 4.6(a). In this case, a value for k of 4 was again used, producing an image that is a much crisper version of the corresponding degraded image. Also note that details, such as the particular shape of the object in the picture, are enhanced. The noise in the background is not greater than that in the original, which means that the desired result of contrast enhancement was achieved without a significant noise increase.

The results of processing the image in Figure 4.6(b) are shown in Figure 4.8(b). In this case a value of k=5 was used, and the contrast of the two objects in the picture are increased with practically no increase in the noise.



(a)



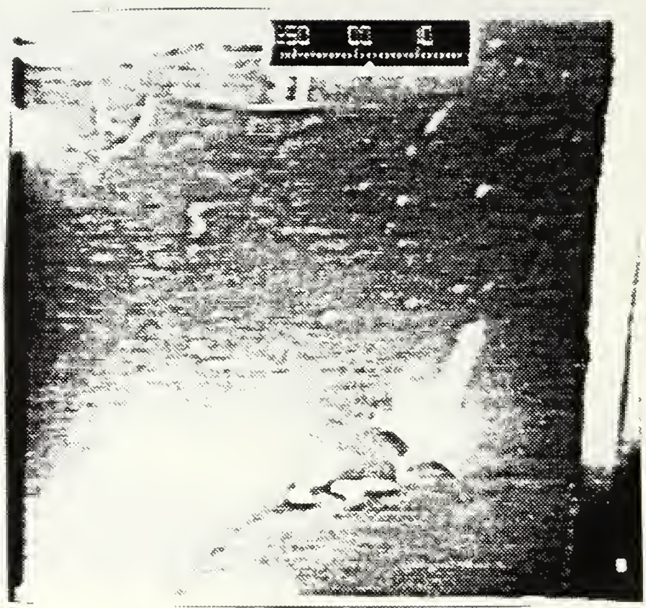
(b)

Figure 4.5. Original Degraded Images

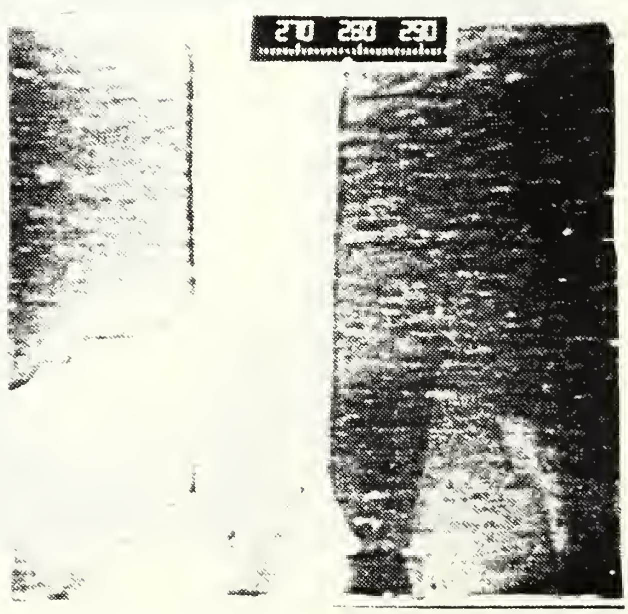
(a)

(b)

Figure 4.6. Original Degraded Images

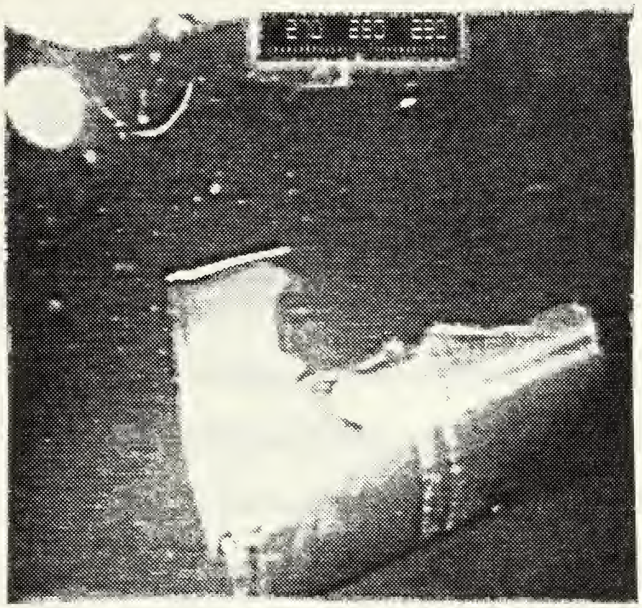


(a)

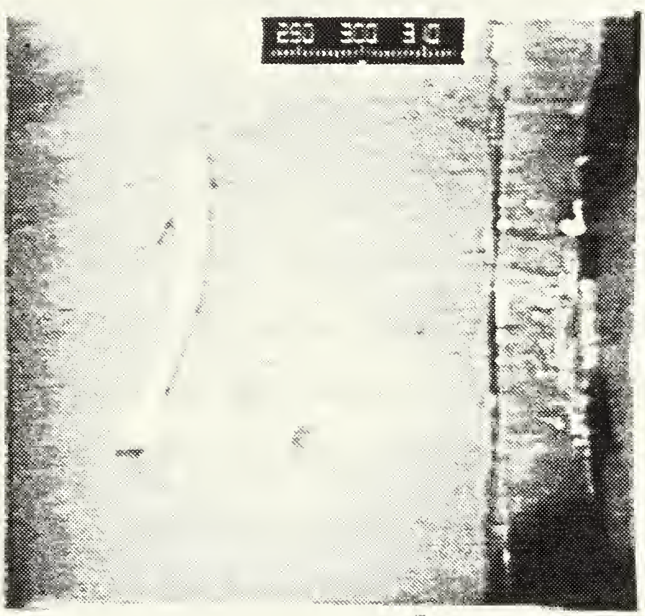


(b)

Figure 4.7. Images in Figure 4.5 Processed with the Contrast Enhancement by Use of Local Statistics Algorithm



(a)



(b)

Figure 4.8. Images in Figure 4.6 Processed with The Contrast Enhancement by Use of Local Statistics Algorithm

V. CONCLUSIONS

Restoration of images degraded by turbid water was considered in this thesis. Previous work on the subject included the use of an adaptive filtering algorithm for enhancing the local contrast. This algorithm was applied to images characterized by poor contrast due to the turbid water viewing conditions. The technique yielded good contrast enhancement, but tended to accentuate the noise.

In our work some techniques for noise smoothing were used as post-processors to the contrast enhancement operation.

The specific techniques used were short space spectral subtraction, noise filtering by use of local statistics, and median filtering. The spectral subtraction algorithm yields very good noise reduction, but tends to introduce some signal degradation. The filtering technique that uses local statistics -local mean and variance- produced noise smoothing in the background. This technique was found to be very sensitive to the size of the window in which the local statistics were evaluated. Median Filtering did not eliminate the noise completely, but the remaining noise was of much lower spatial frequency than the original noise and thus seemed less objectionable. Better noise filtering was achieved when the degraded image was median-filtered iteratively until some signal degradation began to appear.

As an alternative to the adaptive filtering algorithm, an algorithm for contrast enhancement by use of local statistics was used. This particular technique yielded very good results without significant noise increase.

All processes implemented in this thesis proved to be quite computationally intensive. This was especially true for the techniques that require the evaluation of local statistics. The ultimate goal for the problem addressed in this and previous work, is to perform the enhancement in real time. This suggests a natural continuation for the work done so far on this topic, in order to implement the algorithms for enhancement in real time.

APPENDIX

COMPUTER PROGRAMS

```
          PROGRAM SPATSUB
C
C
C         THIS PROGRAM IS THE IMPLEMENTATION OF THE SHORT SPACE
C         SPECTRAL SUBTRACTION ALGORITHM
C
C
C

          BYTE BYTEIMG(512,512)
          CHARACTER* 1 COEN
          CHARACTER*40 INFILENAME,OUTFILENAME
          INTEGER*4 IMGBUFF(512,512), INTARRAY(512,512)

C         *** INPUT FILENAMES : INPUT & OUTPUT ***
          TYPE 10
10        FORMAT (' INPUT FILENAME => ',$)
          ACCEPT 15, INFILENAME
15         FORMAT (A40)

C         ACCEPT 67, IWY
67        FORMAT (I2)
          TYPE 68
68         FORMAT (' COLUMN SIZE OF WINDOW => ',$)
          IWX =16
C         ACCEPT 69, IWX
69        FORMAT (I2)


          IERR = 0

C         *** INPUT ORIGINAL IMAGE ***

          CALL INPUTIMG (INFILENAME,BYTEIMG, IMGX, IMGY)

C         *** CHANGE IMAGE FROM BYTE TYPE TO INTEGER TYPE ***

          CALL BYTE_TO_INTEGER (BYTEIMG, IMGBUFF, IMGX, IMGY, IERR)
          IF (IERRR .EQ. 1) GO TO 100

C         *** PERFORM THE SPATIAL SUBSTRACTION ***
```

```
      CALL CORTO (IMGBUFF, INTARRAY, IMGX, IMGY, IWY, IERR)

      IF (IERR .EQ. 1 ) GO TO 100

C     *** CHANGE IMAGE FROM INTEGER TYPE TO BYTE TYPE ***
      CALL INTEGER_TO_BYTE (INTARRAY, BYTEIMG, IMGX, IMGY,IERR)
      IF (IERR . EQ. 1) GO TO 100

C     *** OUTPUT THE PROCESSED IMAGE ***

      CALL OUTPUTING (OUTFILENAME, BYTEIMG, IMGX, IMGY)

      STOP

      CONTINUE
      TYPE  *,' !! FORCED TO EXIT !! '
      STOP

      END
      SUBROUTINE CORTO (ENTRADA, SALIDA,IMGX,IMGY,IWY,IERR)

C     THIS SUBROUTINE PERFORMS THE SPATIAL SUBTRACTION ALGORITHN
C     USING THE SHORT SPACE IMPLEMENTATION TECHNIQUE FOR IMAGE
C     RESTORATION


      INTEGER*4 ENTRADA (IMGX,IMGY), SALIDA(IMGX,IMGY), INTE(32,32)

      REAL*4  AR(32,32),AI(32,32),BR(32,32),BI(32,32),
     +  ST(64),  CT(64),  LBR(64), BRR(32.32), WINDOW(32,32),
     +  RENTRADA(512,512), IMGWIN(32,32), IARRAY(32,32), MAG(32,32),
     +  PHAS(32,32),PSD(32,32)

      DO I  = 1,IMGX
      DO J = 1,IMGY
                RENTRADA  (I,J) = FLOATJ(ENTRADA(I,J))
      END DO
      END DO

      TYPE *,'INTEGER IMAGE,CONVERTED TO REAL DATA'

C     CALL MCHECK(RENTRADA,IMGX)

C     ***  BUILD UP THE WINDOW FUNCTION ***

      CALL BLTWINDOW (16,WINDOW)

C     TYPE *,'WINDOW FUNCTION BUILT'

C     CALL MCHECK(WINDOW,32)
```

```
           AK = 0.0

           DO I = 1,32
           DO J =1,32
                  AK = AK+(WINDOW(I,J)**2
           END DO
           END DO

           TYPE *,'AK=>... ',AK

C          *** INITIALIZE OUTPUT ARRAY ***

           DO L = 1,IMG
           DO M = 1,IMGY
                  SALIDA (L,M) = 0
           END DO
           END DO

C          TYPE *,'OUTPUT ARRAY INITIALIZED'

C          CALL MCHECK(SALIDA,IMGX)

C          *** ESTIMATE POWER SPECTRUM DENSITY OF THE NOISE ***

           CALL ESTINOISE(RENTRADA,PSD)

C          TYPE *,'POWER SPECTRUM DENSITY OF THE NOISE ESTIMATED'

C          CALL MCHECK(PSD,32)

C          *** MAIN PROCESS SECTION ***

C          INSERT LIMITS HERE

           DO I = 1,31
           DO J = 1,31
                  LL = ((I-1)*16)+1
                  MM = ((J-1)*16)+1
                  DO L = 1,32
                  DO M = 1,32
                        IA = LL+L-1
                        B = MM+M-1
C                       IF(IA .GE. 16 .AND IA .LE. 112 .AND.
C     +                  IB .GE. 16 .AND. IB .LE. 112) THEN
                              IMGWIN(L,M)=RENTRADA(IA,IB)
                              IARRAY(L,M)=WINDOW(L,M)*IMGWIN(L,M)

C                       ELSE IF(IA .GE.16 .AND. IA .LE. 112) THEN
C                             IARRAY(L,M)= 0.5*RENTRADA(IA,IB)
C                       ELSE IF(IB .GE. 16 .AND. IB .LE. 112) THEN
C                             IARRAY(L,M)=0.5*RENTRADA(IA,IB)
```

```
C                   ELSE
C                           IARRAY (L,M)=RENTRADA(IA,IB)
C
C                   END IF
             END DO
             END DO

C            TYPE *,'DATA ARRAY HAS BEEN WINDOWED'

C            CALL MCHECK (IARRAY,32)

C            *** TRANSFORM DATA ARRAY INTO COMPLEX FORMAT ***

             DO L=1,32
             DO M=1,32
                    AI(L,M) = 0.
                    AR(L,M) = IARRAY(L,M)
             END DO
             END DO


C            TYPE*,'CHECK REAL PART'

C            CALL MCHECK (AR,32)

C            TYPE *,'CHECK IMAGINARY PART (ZEROS)'
C            CALL MCHECK (AI,32)

C            *** EVALUATE 2-D FFT OF DATA IMAGE ***

             CALL   FFTS2(AR,AI,BR,BI,32,32,ST,CT,LBR,64,2,JERR)
C            CALL MCHECK (BR,32)
             DO IX  = 1,32
             DO IY = 1,32
                    BRR(IX,IY)=CABS   (CMPLX(BR(IX,IY),BI(IX,IY)))
             END DO
             END DO

C            TYPE *,'FFT DONE,CHECK MAGNITUDE'

C            CALL MCHECK (BRR,32)

C            CALL DISPECT (BRR)

C            ***PERFORFM SPATIAL SUBSTRACTION ***

             ALFA = 1.5
C            PSD = 40.0
             DO IX = 1,32
             DO IY = 1,32
                    BR(IX,IY) = ((BRR(IX,IY))**2)  -  ((ALFA/AK) * ((PSD(IX,IY) **2)))
```

```
                  IF (BR(IX,IY) .LT. .0.) THEN
                          MAG (IX,IY) = 0
                  ELSE
                          MAG(IX,IY) = SQRT(BR(IX,IY))
                  END IF
          END DO
          END DO

C         TYPE  *,'SPATIAL SUBT.DONE CHECK RESULT'

C         CALL MCHECK (MAG)

C         *** EVAL. MAGNITUDE OF INV. FFT ***

          DO IX= 1,32
          DO IY= 1,32
                  BR(IX,IY) = MAG(IX,IY)*COS(PHAS(IX,IY)
                  BI (IX,IY) = MAG(IX,IY)*SIN(PHAS(IX,IY)
          END DO
          END DO

          CALL  FFTS2(BR,BI,AR,AI,32,32,ST,CT,LBR,64,-2,JERR)

C         TYPE*,'INV. FFT DONE, CHECK REAL PART'
C         CALL MCHECK(AR)

C         TYPE*,'CHECK IMAGINARY PART (ZEROS) '
C         CALL MCHECK(AI)

C         *** BACK TO INTEGER FORMAT ***

          DO IX = 1,32
          DO IY = 1,32
                  INTE(IX,IY) = JINT(AR(IX,IY))
          END DO
          END DO

C         TYPE*,'INTEGER CONVERSION DONE,CHECK IT'
C         CALL MCHECK (INTE)

C         *** BUILD UP THE OUTPUT ARRAY ***

          DO IX = 1,32
          DO IY = 1.32
                  LL =((I-1)*16)+1
                  MM((J-1)*16)+1
                  IXX =LL+IX-1
                  IYY =MM+IY-1
                  SALIDA (IXX,IYY) = SALIDA(IXX,IYY)+INTE(IX,IY)
C                 TYPE*,'IXX AND IYY =>',IXX,IYY
```

```
                END DO
                END DO

        END DO !! END DO J
        END DO !! END DO I

100     TYPE*,'MAIN PROCESS DONE'

C       CALL  TRUNCATE(SALIDA,IMGS,IMGY)
        CALL  SCALE_ADJUST(SALIDA,IMGX,IMGY,0.0,255.0)
        TYPE  *,'SPATIAL PROCESS DONE'

        RETURN

        END



        SUBROUTINE BLTWINDOW(IWY,NIN2)
        REAL*4 NI(32),  NIN2(32,32), N2(32,32)

        L = 0
        PI = 4*ATAN (1.)
        DO I + 1,32
                II = I-1
                N1(I) = 0.5 - (0.5*COS(2*PI*II/31))
                N2(I) = N1(I)
C               TYPE *, N1(I), N2(I)
C               IF (I .GT. 16) THEN
C                       L = L+1
C               N1(I) = I - (2*L)
C               ELSE

C               N1(I) = I
C               END IF
        END DO

        L = 0
        DO I = 1 , 32
        DO J = 1, 32
C               IF (J .GT. 16) THEN
C                       L = L +1
C                       NIN2(I,J) = NI(I) * (J-2* 1)
C               ELSE
C                       NIN2(I,J) = NI(I) * J
C               END IF
                NIN2(I,J) = N1(I)*N2(J)
C       TYPE *, NIN2(I,J)
        END DO
        L = 0
```

```
      END DO

      RETURN

      END


      SUBROUTINE ESTINOISE (RENTRADA,PSD)

C     THIS ROUTINE ESTIMATES THE POWER  SPECTRAL DENSITY OF
C     THE NOISE IN AN IMAGE, BY TAKING THE MAGNITUDE (SQUARED)
C     OF THE DISCRETE SPACE FOURIER TRANSFORM OF A (NOISY) PART
C     OF THE BACKGROUND.

C     TO CHOOSE APPROPIATE REGION, SET PARAMETERS I AND J
C     ACCORDINGLY.

      REAL*4 RENTRADA(512,512), PSD(32,32), AR(32,32), AI(32,32),
      BR(32,32), BI(32,32), ST(32,32), CT(32,32), LBR(64)

C     *** TAKE A PORTION OF THE IMAGE BACKGROUND ***

      J=80
      K=80
      DO L=1,32
      DO M=1,32
            LL=L+J
            MM=M+K
            AR(L,M)=RENTRADA(LL,MM)
      END DO
      END DO

C     *** INITIALIZE IMAGINARY PART OF DATA ***

      DO L=1,32
      DO M=1,32
            AI(L,M)=0.
      END DO
      END DO

C     *** EVALUATE MAGNITUDE OF 2-D FFT ***

      CALL  FFTS2(AR,AI,BR,BI,32,32,ST,CT,LBR,64,2,JERR)

      DO L=1,32
      DO M=1,32
            PSD(L,M)=CABS(CMPLX(BR(L,M),BI(L,M)))
      END DO
```

```
      END DO

      RETURN

      END


      SUBROUTINE  INPUTIMG(INFILENAME,BYTEIMG,IMGX,IMGY)

C     READ IN INPUT IMAGE DATA IN BYTE TYPE
C     ASSUME INPUT DATA FILE IS A DIRECT ACCESS FILE
      BYTE  BYTEIMG(IMGY,IMGX)
      CHARACTER*40  INFILENAME


      OPEN(UNIT=2, FILE=INFILENAME, STATUS='OLD', ACCESS='DIRECT' ,
     +     RECORDTYPE='FIXED')


      DO I=1,IMGY
      READ(2'I) (BYTEIMG(I, J), J=1, IMGX)
      END DO

      TYPE *, ' * INPUT IMAGE HAS BEEN READ '

      CLOSE(2)

      RETURN
      END


      SUBROUTINE BYTE_TO_INTEGER (BYTEDATA, INTDATA, IX, IY, IERR)
C
C     THIS PROGRAM CHANGE DATA (USUALLY 2D IMAGE DATA) IN THE BYTE DATA
C     TYPE INTO THE INTEGER TYPE
C

      BYTE      BYTEDATA (IY, IX)
      INTEGER*4 INTDATA (IY, IX)

      DO I = 1, IY
      DO J = 1, IX

      IF (BYTEDATA (I,J) .GE. -128
     +     .AND. BYTEDATA (I,J) .LT. 0) THEN
      INTDATA(I,J) = BYTEDATA (I,J) + 256
      ELSE IF (BYTEDATA(I,J) .GE.0
     +     .AND. BYTEDATA(I,J) .LE. 127) THEN
      INTDATA(I,J) = BYTEDATA(I,J)
      ELSE
```

```
      TYPE*,'%DATA OUT OF RANGE! ROW, COL, DATA => ',
     +      I, J, BYTEDATA(I,J)
      IERR = 1
      GO TO 100
      ENDIF

      END DO
      END DO

      TYPE*, 'BYTE TO INTEGER CONVERSION HAS BEEN DONE'

100   CONTINUE

      RETURN
      END
```

SUBROUTINE INTEGER_TO_BYTE(INTDATA, BYTEDATA, IX, IY, IERR)

```
C
C     THIS PROGRAM CHANGE DATA (USUALLY 2D IMAGE DATA) IN THE BYTE DATA
C     TYPE INTO THE INTEGER TYPE
C

      BYTE BYTEDATA(IX, IY)
      INTEGER*4 INTDATA(IX, IY)

      DO I=1,IX
      DO J=1,IY

      IF (INTDATA(I,J) .LT. 0 .OR.  INTDATA(I,J) .GT. 255) THEN TYPE*,'%
      DATA OUTOF RANGE! ROW,COL,DATA => ',
     +      I, J, INTDATA(I,J)
      IERR = 1
      GO TO 100
      ELSE IF (INTDATA(I,J) .1E. 127) THEN
      BYTEDATA(I,J) = INTDATA(I,J)
      ELSE
      BYTEDATA(I,J) = INTDATA(I,J) - 256
      ENDIF

      END DO
      END DO

      TYPE *,' INTEGER TO BYTE CONVERSION HAS BEEN DONE'

100   CONTINUE
      RETURN
      END
```

```
      PROGRAM LOC_STAT

C           THIS PROGRAM IS THE IMPLEMENTATION OF THE IMAGE ENHANCEMENT AND
C           NOISE FILTERING ALGORITHM
C
C
      BYTE BYTEIMG (512,512)
      CHARACTER*1 COEN
      CHARACTER*40 INFILENAME, OUTFILENAME
      INTEGER*2 IMGBUFF(512,512), LAVEIMG(512,512), LOCMEAN(512,512),
     - INTARRAY(512,512)

C           *** INPUT FILENAMES : INPUT & OUTPUT ***
      TYPE 10
10    FORMAT (' INPUT FILENAME => ',$)
      ACCEPT 15, INFILENAME
15    FORMAT (A40)
      TYPE 20
20    FORMAT (' OUTPUT FILENAME => ',%)
      ACCEPT 25, OUTFILENAME
25    FORMAT (A40)

C           *** INPUT ORIGINAL IMAGE SIZE ***

      TYPE 50
50     FORMAT (' ROW SIZE OF IMAGE => ',$)
      ACCEPT 55, IMGY
55    FORMAT (I4)
      TYPE 60
60    FORMAT (' COLUMN SIZE OF IMAGE => ',$)
      ACCEPT 65, IMGX
65    FORMAT 9I4)

C           *** INPUT WINDOW SIZE ***

      TYPE 66
66      FORMAT (' ROW SIZE OF WINDOW => ',$)
      ACCEPT 67, IWY
67    FORMAT (I2)
      TYPE 68
68    FORMAT (' COLUMN SIZE OF WINDOW => ',$)
      ACCEPT 69, IWX
69    FORMAT (I2)

C           ***SELECT CONTRAST ENHANCEMENT OR NOISE FILTERING ***

      TYPE 70
70    FORMAT ( 'CONTRAST/NOISEFIL ? (C/N) => ', $)
```

```
      ACCEPT 75, CE
75    FORMAT (A1)


      IERR = 0

C     ***INPUT ORIGINAL IMAGE ***

      CALL INPUTIMG (INFILENAME, BYTEIMG, IMGX, IMGY)

C     ***CHANGE IMAGE FROM BYTE TYPE INTEGER TYPE***

      CALL BYTE_TO_INTEGER ( BYTEIMG, IMGBUFF, IMGX, IMGY, IERR)
      IF (IERR .EQ.1) GOTO 100

C     *** CALCULATE LOCAL MEAN ARRAY ***

      CALL LOCAL_MEAN(IMGBUFF, LOCMEAN, IMGX, IMGY, IWX, IWY, IERR)
      IF(IERR .EQ.1 ) GOTO 100

      IF (CE .EQ. 'C' .OR. CE .EQ. 'c') THEN

C     *** INPUT K = STANDARD DEVIATION RATIO ***
      TYPE 80
80    FORMAT (' STAND. DEV. RATIO K=>', $)
      ACCEPT 85, K
85    FORMAT (I4)
      CALL ENHACONT (IMGBUFF, LOCMEAN, INTARRAY, IMGX, IMGY, K, IERR)
      ELSE

C     ***INPUT NV = NOISE VARIANCE ***
      TYPE 90
90     FORMAT (' NOISE VARIANCE NV => ', $)

      ACCEPT 95
95    FORMAT (I4)
      CALL NOISE_FILT ( IMGBUFF, LOCMEAN, INTARRAY, NV, IMGS, IMGY, IERR)
      ENDIF

      SCLMIN = 0.0
      SCLMAX = 255.0
C     CALL SCALE_ADJUST (INTARRAY, IMGX, IMGY, SCLMAX, SCLMIN)
      CALL TRUNCATE (INTARRAY,IMGX,IMGY)

C     ***CHANGE IMAGE FROM INTEGER TYPE TO BYTE TYPE ***

      CALL INTEGER_TO_BYTE ( INTARRAY, BYTEIMG, IMGX, IMGY, IERR )
      IF (IERR .EQ. 1) GOTO 100

C     ***OUTPUT THE PROCESSED IMAGE ***
```

```
          CALL OUTPUTIMG ( OUTFILENAME, BYTEIMG, IMGX, IMGY )

          STOP
100       CONTINUE
          TYPE *,' !! FORCED TO EXIT !! '
          STOP

          END
```

SUBROUTINE ENHACONT (IMGBUFF,LOCMEAN,INTARRAY,IMGX,IMGY,K,IERR)

```
C         THIS SUBROUTINE IMPLEMENTS THE CONTRAST ENHANCEMENT BY USE
C         OF LOCAL STATISTICS ALGORITHM.
C

C
          INTEGER*2 IMGBUFF(IMGY,IMGX) , LOCMEAN(IMGY,IMGX) ,
          INTARRAY(IMGY,IMGX)

C         *** EVALUATE LOCAL CONTRAST AND MODIFY IT WITH K ***

          DO I = 1 IMGY
          DO J = 1,IMGX
          INTARRAY(I,J) = K*(IMGBUFF(I,J) - LOCMEAN(I,J))
          END DO
          END DO

C         *** ADD MODIFIED LOC. CONTRAST AND LOCAL MEAN ***

          DO I = 1,IMGY
          DO J = 1,IMGX
          INTARRAY (I,J) = INTARRAY (I,J) + LOCMEAN (I,J)
          END DO
          END DO

C         *** CORRECT ANY OVERFLOW ***

          DO I = 1,IMGY
          DO J = 1,IMGX
          INTARRAY(I,J) = K*(IMGBUFF(I,J) - LOCMEAN(I,J) )
          END DO
          END DO

C         *** ADD MODIFIED LOC. CONTRAST AND LOCAL MEAN ***

          DO I = 1,IMGY
          DO J = 1,IMGX
          INTARRAY (I,J) = INTARRAY (I,J) + LOCMEAN (I,J)
          END DO
```

```
      END DO

C     *** CORRECT ANY OVERFLOW ***

      SCLMIN = 0.0
      SCLMAX = 255.0
C     CALL SCALE_ADJUST (INTARRAY, IMGX, IMGY, SCLMAX, SCLMIN)

      RETURN
      END


      SUBROUTINE  NOISE_FILT(IMGBUFF,LOCMEAN,INTARRAY,NV,IMGX,
     + IMGY,IERR)


C     THIS SUBROUTINE IS THE IMPLEMENTATION OF THE NOISE FILTERING
C     ALGORITHM, CALLED BY THE NOISE FILTERING AND CONTRAST ENHANCEMENT
C     BY USE OF LOCAL STATISTICS PROGRAM.
C
C     REFERENCES:
C
C
      INTEGER*2  IMGBUFF(IMGY,IMGX),  LOCMEAN(IMGY,IMGX),
     + INTARRAY(IMGY,IMGX)
      REAL*8 AK(128,128) ,Q(128,128)

C     *** EVALUATE THE VARIANCE OF THE IMPUT IMAGE ***

      TYPE  *,'CHECK  IMGBUFF  ARRAY'
C     CALL  MCHECK(IMGBUFF)

      TYPE  *,'CHECK  LOCMEAN  ARRAY '
C     CALL  MCHECK(LOCMEAN)
      DO I  =  1,IMGY
      DO J  =  1,IMGX
      IDIF  =  IMGBUFF(I,J)  -  LOCMEAN(I,J)
C     TYPE  16, IDIF
16    FORMAT  (' DIF => ', I10)
      Q(I,J) = (IDIF**2) - NV
C     TYPE  17, Q(I,J)
17     FORMAT(' Q => '.F8.2)
      END DO
      END DO

      TYPE *,' CHECK THE Q ARRAY '
C     CALL MRCHECK(Q)
C     *** EVALUATE THE GAIN AK ***

      DO I= 1,IMGY
      DO J = 1, IMGX
```

```
                DEN = Q(I,J) + NV
                IF ( DEN .EQ. 0.0) THEN
                        DEN = 1.0
                ENDIF
                AK(I,J) = (Q(I,J) / DEN)
        END DO
        END DO

C       *** ESTIMATE THE RESTORED IMAGE ***

        DO I = 1, IMGY
        DO J = 1, IMGX
                DIF  =  AK(I,J)*(IMGBUFF(I,J)  -LOCMEAN(I,J))
                INTARRAY(I,J) = INT(LOCMEAN(I,J) + DIF)
        END DO
        END DO

        RETURN
        END


        SUBROUTINE SCALE_ADJUST (IMGBUFF, IMGX, IMGY, SCLMAX, SCLMIN)

        INTEGER*4 IMGBUFF (IMGX, IMGY)

        MAXVAL = IMGBUFF (1,1)
        MINVAL = IMGBUFF (1,1)

        DO I =1, IMGY
        DO J = 1, IMGX

                IF (IMGBUFF(I,J) .GT. MAXVAL) THEN
                        MAXVAL = IMGBUFF(I,J)

                ENDIF

                IF (IMGBUFF(I,J) .1T. MINVAL) THEN
                        MINVAL = IMGBUFF(I,J)
                ENDIF

        END DO
        END DO

C       TYPE 10, MAXVAL, MINVAL
C10     FORMAT('MAX &MIN VALUES =>' 2I8)
        TYPE *,'MAXVAL= ',MAXVAL, 'MINVAAL= ',MINVAL

        INTVAL = MAXVAL - MINVAL

        SLOPE = (SCLMAX - SCLMIN)/REAL (INTVAL)
```

```
      DO I = 1, IMGY
      DO J = 1, IMGX

            IF (IMGBUFF(I,J) .EQ. MINVAL) THEN
                  IMGBUFF (I,J) = SCLMIN

            ELSE IF (IMGBUFF(I,J) .EQ. MAXVAL) THEN
                  IMGBUFF (I,J) = SCLMAX
            ELSE
                  IDIF = IMGBUFF (I,J) - MINVAL
                  IMGBUFF(I,J) = INT(S) OPE*IDIF) + INT(SCLMIN
            ENDIF

      END DO
      END DO

      RETURN
      END
```

SUBROUTINE TRUNCATE(IMGBUFF, IMGX, IMGY)

```
C     TRUNCATE IMAGE GRAY VALUES INTO 0-255

      INTEGER*4 IMGBUFF(IMGY, IMGX)

      DO I = 1, IMGY
      DO J = 1, IMGX
            IF (IMGBUFF(I,J) .LT. 0) IMGBUFF(I,J) = 0
            IF (IMGBUFF(I,J) .GT. 255) IMGBUFF(1,J) = 255
      END DO
      END DO

      RETURN
      END
```

```
            PROGRAM MEDIFILT

C           THIS PROGRAM IS THE IMPLEMENTATION OF THE MEDIAN
C           FILTERING IN NOISE REDUCTION


            BYTE BYTEIMG(512, 512)
            CHARACTER*40 INFILENAME, OUTFILENAME
            INTEGER*4 IMGBUFF(512,512), IHST(256), INTARRAY(512, 512)


C           *** INPUT FILENAMES : INPUT & OUTPUT ***
            TYPE 10
10          FORMAT ('INPUT FILENAME => ',$)
            ACCEPT 15, INFILENAME
            FORMAT (A40)
            TYPE 20
20           FORMAT ('OUTPUT FILENAME  => ',$)
            ACCEPT 25, OUTFILENAME
25          FORMAT (A40)

C           *** INPUT ORIGINAL IMAGE SIZE ***

C           TYPE 50
50          FORMAT ('ROW SIZE OF IMAGE => ',$)
            IMGY=512
C           ACCEPT 55, IMGY
55          FORMAT (I4)
C           TYPE 60
60          FORMAT (' COLUMN SIZE OF IMAGE => ', $)
            IMGX = 512
C           ACCEPT 65, IMGX
65          FORMAT (I4)

C           TYPE *,IMGX,IMGY

            IERR = 0

C           *** INPUT ORIGINAL IMAGE ***

            CALL IMPUTIMG (INFILENAME,BYTEIMG, IMGX, IMGY)

C           *** CHANGE IMAGE FROM BYTE TYPE TO INTEGER TYPE ***

            CALL BYTE_TO_INTEGER (BYTEIMG, IMGBUFF, IMGX, IMGY, IERR)
            IF (IERR .EQ. 1 ) GO TO 100
C           ***  PERFORM THE MEDIAN FILTERING ***
            DO I = 1, 20
            CALL MEDI (IMGBUFF, INTARRAY, IMGX, IMGY, 5, 5, IHST, 256)
                    DO IX = 1, IMGX
```

```
                DO IY = 1, IMGY
                        IMGBUFF(IX,IY) = INTARRAY(IX, IY)
                END DO
                END DO
        END DO
C       *** CHANGE IMAGE FROM INTEGER TYPE TO BYTE TYPE ***
C       TYPE *, IMGX, IMGY
        CALL INTEGER_TO_BYTE (INTARRAY, BYTEIMG, IMGX, IMGY, IERR)
        IF (IERR .EQ. 1) GO TO 100

C       *** OUTPUT THE PROCESSED IMAGE ***

        CALL OUTPUTIMG (OUTFILENAME, BYTEIMG, IMGX, IMGY)

        STOP

100     CONTINUE
        TYPE *,' !! FORCED TO EXIT !!'
        STOP

        END
```

LIST OF REFERENCES

1. Gonzalez, R., Digital Image Processing, Addison-Wesley, 1977.

2. Pratt, W. K., Digital Image Processing, John Wiley & Sons, Inc., 1978.

3. Chellappa, R. and Sawchuk, A. A., Digital Image Processing and Analysis: Volume 1: Digital Image Processing, IEEE Computer Society Press, 1985.

4. Peli, T. and Lim, J. S., "Adaptive Filtering for Image Enhancement," Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing, March 30, 31 & April 1, 1981.

5. Lincoln Laboratory, M.I.T., TN 1979-17, Image Restoration by Short Space Spectral Subtraction, by J.S. Lim, 1979.

6. Lee, J. "Digital Image Enhancement and Noise Filtering by Use of Local Statistics," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, Number 2, pp 165-168, 1980.

7. Tukey, J. W., Exploratory Data Analysis, Addison-Wesley, 1971.

8. Narendra, P. M., "A Separable Median Filter for Image Noise Smoothing," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, Number 1, pp 20-29, January 1981.

9. Wendt, P.D., Coyle, E.J., and Gallagher, N.C., "Some Convergence Properties of Median Filters," IEEE Transactions on Circuits and Systems, Vol. CAS-33, Number 3, March 1986.

10. Gallagher, N.C. and Wise, G.L., "A theoretical Analysis of the Properties of Median Filters," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-29, pp 1136-1141, 1981.

11. Huang, T.S., Yang, G.J., and Tang, G.Y., "A Fast Two-Dimensional Median Filtering Algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-27, Number 1, 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Professor C. W. Therrien, Code 62Ti Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	3
5.	Professor P. H. Moose, Code 62Me Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
6.	Sr. Almirante Comandante de la Armada Comando Armada Nacional CAN, Avenida ElDorado Bogotá, D.E. Colombia	2
7.	Sr. Contralmirante Director Escuela Naval Escuela Naval "Almirante Padilla" Cartagena Colombia	2

8. Mr. Richard Evans, Code 7021 1
NUWES Code 7021
Naval Undersea Weapons Engineering Station
Keyport, Washington 98345

9. Ms. Tami Peli 1
Gr. 21
M.I.T. Lincoln Laboratory
Lexington, Massachusetts 02173-0073

10. Professor Jae Lim 1
Department Elec. Eng. & Computer Science
Room 36-653
M.I.T.
Cambridge, Massachusetts 02139

11. Professor O. B. Wilson, Code 61Wl 1
Department of Physics
Naval Postgraduate School
Monterey, California 93943

12. Ms. Atsuko Keiyu , Code 62 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, California 93943

13. Teniente de Navío Jorge A. Franco L. 2
Escuela Naval "Almirante Padilla"
Cartagena
Colombia

221006
Thesis
F754 Franco
c.1 Enhancement of video images degraded by turbid water.